

Thema der Ausgabe

Erweiterte Rechte in Citrix-Umgebungen

Marc Ruef

Schwierigkeitsgrad



Zentralisierte Lösungen wie Citrix sind auf dem Vormarsch. Durch das Freigeben dedizierter Applikationen lässt sich über das Netzwerk eine virtuelle Arbeitsumgebung realisieren. Dabei werden zu Gunsten der Sicherheit die Rechte des Benutzers möglichst limitiert. Derartige Multiuser-Umgebungen sind jedoch enormen Risiken ausgesetzt.

Zu Beginn des Zeitalters der praktischen Informatik fielen die Kosten für eine entsprechende Installation enorm hoch aus. Dies führte dazu, dass sich zentralisierte Host-Lösungen etablieren konnten. Durch die einzelnen Systeme konnte für mehrere Benutzer ein Dienst bereitgestellt werden. Die Host-Systeme fungierten damit als Multiuser-Umgebungen, die den Anwendern Ressourcen bereitstellen konnten. Um einen Zugriff umsetzen zu können, war lediglich eine einmalige Anschaffung vergleichsweise kostengünstiger Hardware erforderlich.

Mit der Popularisierung der elektronischen Datenverarbeitung fielen auch die Preise für Hardware. Der Kostendruck auf der Nutzerseite war nicht mehr gegeben, weshalb eine gewisse Dezentralisierung Einzug hielt. Immermehr wurden die Clients aufgerüstet und damit die zentralisierten Host-Lösungen uninteressant. Die Benutzer konnten auf ihren dedizierten Systemen den Arbeiten nachgehen und waren damit nicht mehr von den Hosts abhängig.

Das Thema Sicherheit hat in den letzten Jahren in dieser Hinsicht wieder eine Umwälzung in die Wege geleitet. Zunehmends wird auf zentralisierte Lösungen zurückgeschwenkt. Dadurch lassen sich zentralisiert administrative Eingriffe vornehmen. Ebenso ist die Überwachung einfacher zu bewerkstelligen.

Ein Produkt, das sich in den letzten Jahren zum Marktführer entwickelt hat, ist die kommerzielle Lösung von Citrix (*http://www.citrix.com*). Damit wird ein Zusatz für moderne Windows-Systeme geboten, um diese um eine echte Multiuser-Funktionalität, die über ein Netzwerk genutzt werden kann (ähnlich X11 aus dem Unix-Umfeld), zu erweitern. Verschiedene Benutzer

In diesem Artikel erfahren Sie...

- wie die Architektur von Citrix aussieht;
- welche Möglichkeiten der Rechteausweitung gegeben sind;
- wie sich Attacken durchführen;
- und wie sich solche entdecken lassen.

Was Sie vorher wissen/können sollten...

- Arbeiten mit Windows;
- Grundlagen der Computersicherheit.

können durch einen Citrix-Client zeitgleich auf den Server arbeiten. Dabei werden ihnen Applikationen zur Verfügung gestellt, die in ihrem spezifischen Benutzerkontext laufen.

Zentralisierte Lösungen wie Citrix sind aber auch für Angreifer von Interesse. Sind in einem dezentralisierten Netzwerk die verschiedenen Elemente als Angriffsziele anzusehen, wird bei einer zentralisierten Lösung der Fokus auf eben jene gelegt. Durch das Angreifen dieser wird es möglich, durch einen Schlag sämtliche Clients bzw. die Daten der einzelnen Benutzer unter Kontrolle zu bringen. Damit ist ein *Common Point of Failure* gegeben, der die Komplexität von Angriffen von heterogenen Umgebungen zunichte macht.

Im Zusammenhang mit Citrix sind Administratoren deshalb darum bemüht, einen möglichst reduzierten und in sich geschlossenen Benutzerkontext anzubieten. Dabei werden ausschliesslich einzelne Applikationen, die vom Anwender erfordert sind, freigegeben (engl. *published*). Eine publizierte Anwendung sollte nicht verlassen werden können, um nicht Gefahr zu laufen, Einblick in geschützte Bereiche ermöglichen zu können. Aber genau dies wird ein Angreifer versuchen wollen.

Das Dateisystem mit Auto-Complete erkunden

In der Regel bekommt ein Citrix-Benutzer die für seine Arbeit erforderlichen Applikationen zugewiesen. Diese bilden das Frontend, mit dem der anstehenden Tätigkeit nachgegangen werden kann. Wird beispielsweise in einer sicherheitsbewussten Organisation auf den Direktzugriff auf das Internet über die Arbeitsplatzrechner verzichtet (z.B. wegen Virengefahr), kann durch Citrix ein sicherer und dedizierter Internet-Zugang bereitgestellt werden. Die zentralisierte Administration dessen ermöglicht es, eigene Richtlinien in Bezug auf Konfigurationen (z.B. kein Download von EXE-Dateien, kein Ausführen von Java ausserhalb des Intranets) und vorinstallierte Applikationen (z.B. einheitliche Browser-Version, aktualisierte Antiviren-Datenbank) zu erzwingen. Die Benutzer verbinden sich also von ihrem Arbeitsplatzrechner auf den Citrix-Server, durch den sie im Internet browsen können. Die lokalen Arbeitsplatzrechner und damit das interne LAN sind durch die Entkoppelung vom Internet vor den bestehenden Gefahren geschützt. Citrix übernimmt in diesem Beispiel die Rolle eines absoluten Application Gateways, wie man es aus dem Firewalling-Bereich kennt.

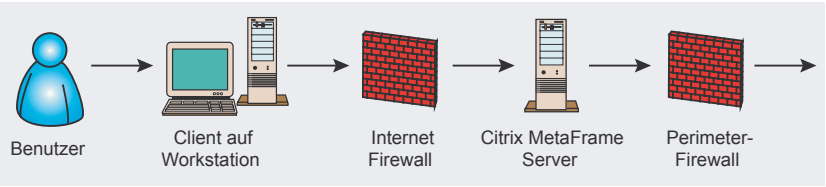


Abbildung 1. *Durch das Verbinden auf einen Citrix MetaFrame Server kann eine Entkoppelung der Clients stattfinden. Die beim Browsen bestehenden Gefahren (z.B. Download von Viren) sind für ihn nicht mehr gegeben*

Für Zwecke des Surfens im Web wird den Benutzern ein Browser (z.B. *Microsoft Internet Explorer*) auf dem Citrix MetaFrame Server zur Verfügung gestellt. Dieser wird nach dem Verbindungsaufbau zum Citrix-Server auf eben jenem gestartet. Die Eingaben und Ausgaben der Software werden über das Netzwerk zum Citrix-Client übertragen. Der Nutzer dieser dedizierten Umgebung kann damit nur auf die ihm zur Verfügung gestellten Ressourcen zugreifen. Ihm sind weder weitere Details zum Betriebssystem noch Zugriffe auf andere Systemkomponenten möglich.

Ein Angreifer, der sich mit legitimen Zugriffsrechten gesegnet sieht, jedoch weitere Rechte erhalten möchte, wird als erstes eine Auswertung des Zielsystems vornehmen [Ruef et al. 2002]. Dabei ist das Zielsystem der Citrix-Server, auf dem schon gewisse Anwendungen gestartet werden konnten. Eines der Ziele solcher Auswertungen lokaler Natur ist das Verständnis für die Struktur des Dateisystems. Diese Information ist von erhöhter Wichtigkeit, da sich nur mit diesem Wissen ein effizienter Zugriff (z.B. das Finden von Passwortdateien) umsetzen lässt.

Sehr viele Anwendungen mit Zugriffsmöglichkeiten auf das Dateisystem, stellen eine dynamische Auto-Complete-Funktion zur Verfügung. Populäres Beispiel ist die Adresszeile des Microsoft Internet Explorers. Dieser kann ebenfalls für das Browsen im lokalen Dateisystem verwendet werden (*explorer.exe*). In der Adresszeile lassen sich deshalb nicht nur URLs im Internet, sondern auch lokale Dateinamen angeben. Um den Zugriff auf letztere zu vereinfachen, wird bei einer Eingabe automatisch eine Auswähl der in Frage kommenden Verzeichnisse und Dateien gewährt.

Wird beispielsweise als Adresse *C:* eingegeben, wird der komplette Inhalt eben jenes Verzeichnisses angezeigt. Ebenso bei Unterverzeichnissen. Wird die Eingabe *C:\ab* vorgenommen, werden sämtliche Dateien und Verzeichnisse der Form *C:\ab** angezeigt (z.B. *C:\abc* und *C:\abc\test.txt*). Von besonderem Interesse ist zum Beispiel der Name des Windows-Verzeichnisses. Wird durch die Einagbe von *C:\WINDOWS* der Inhalt des Verzeichnisses angezeigt (z.B. mit dem Unterverzeichnis *system32*), dann handelt es sich um das Systemverzeichnis. Diese Information lässt beispielsweise Rückschlüsse auf die eingesetzte Windows-Version zu (*Windows 2000* oder *XP/2003*). Desweiteren können damit erweiterte Zugriffe auf sensitive Systemdateien angestrebt werden (z.B. *C:\WINDOWS\ win.ini*).

Vorteil dieser Technik ist, dass keine direkten Dateizugriffe für Objekte umgesetzt werden müssen, wie das anderweitig erforderlich wäre. Stattdessen kann so indirekt ein Datei-Mapping auf einem Datenträger durchgeführt werden. Entsprechende Audit-Einträge in der Ereignisanzeige des Windows-Betriebssystems werden so vermieden und der Angriff bzw. Einbruch vorerst unentdeckt.

Eine Verhaltensweise, die für einen systematisch vorgehenden Angreifer vorteilhaft ist.

Erweiterte Leserechte durch Dialogboxen

Citrix stellt eine multiuser-fähige *Remote-Control* Lösung bereit. Der Benutzer verbindet sich durch seinen ICA-Client auf den Citrix MetaFrame Server. Nach einer klassischen Authentifizierung wird die Benutzerumgebung geladen. In dieser, die parallel instanziert und durch unterschiedliche Benutzer eingesetzt wird, werden dedizierte Ressourcen zur Verfügung gestellt. Um ein Höchstmass an Sicherheit gewährleisten zu können, werden durch sicherheitsbewusste Administratoren lediglich vereinzelte Applikationen anstelle des kompletten Remote-Desktops freigegeben. Die Kommandoeingabe der Zugriffe auf den Citrix-Server erfolgt auf der Client-Seite. Die Tastatur- und Mauskommandos werden über das Netzwerk übertragen und auf dem Server ausgeführt. Wiederum werden die Ausgaben auf dem Server über das Netzwerk dem verbundenen Client zugänglich gemacht.

Abgesehen vom Citrix-Client wird also auf der Client-Seite – im Rahmen des Citrix-Zugriffs - in keinsterweise eine Applikation gestartet. Sämtliche Anwendungen, auf die während der Citrix-Sitzung zugegriffen wird, sind auf dem Citrix-Server gegeben und ausgeführt. Dabei ist die Server-Umgebung darum bemüht, dass dem Client ausschliesslich Zugriff auf jene Daten gewährt wird, die für die Umsetzung der Arbeiten erforderlich sind. Der Angreifer hingegen versucht aus seinem Benutzerkontext auszubrechen.

Eine Vielzahl an Anwendungen kommt mit Dialogboxen daher. Diese werden genutzt, um eine einheitliche Schnittstelle für den Zugriff auf Dateien während dem Öffnen und Speichern von Dateien bereitzustellen (z.B. bei *Microsoft Word*). Hierzu kommt die Komponente *COMDLG32.OCX* (*OLE Control Extension*) zum Einsatz, die durch ihre Flexibelität gegenüber den konventionellen DLL-Bibliotheken überzeugen kann. Auf einfache Weise ist damit das Generieren der entsprechenden Fenster möglich.

Obschon sich durch diese Dialogboxen ausschliesslich eine simple Interaktion mit Datenträgern und ihren Inhalten umsetzen lassen sollte, werden einige zusätzliche Funktionalitäten bereitgestellt. Oftmals wird von der Anwendung, die die Dialogbox generiert hat, ein Dateifilter angewendet (*Files of type*). Dieser wird der Komponente durch die Parameter Filter-Index und Filter gesteuert. Damit will eine Anwendung verhindern, dass versehentlich auf Dateien zugegriffen wird, die nicht interpretiert werden können. Ein Text-Editor könnte mit dem Filter **.txt* darum bemüht sein, lediglich Dateien mit der Endung *.txt* anzuzeigen. Denn das komplexe Format *.doc* mit all seinen Steuerzeichen lässt sich durch die *ASCII*-Interpretation nicht richtig darstellen.

Die Standard-Dialogboxen erlauben es, den Filter der gegenwärtigen Darstellung selber zu bestimmen. Durch das Unterbringen von **.exe* lassen sich zum Beispiel sämtliche ausführbaren *EXE*-Dateien anzeigen. Oder mit den absolutem Filter ** bzw *.** (8.3-Schreibweise) können sämtliche Dateien des gegenwärtigen Verzeichnisses dargestellt werden. Durch diese erste Massnahme ist es möglich über den ersten Tellerrand hinaus zu sehen. In vielen Fällen lassen sich so versteckte Dateien erkennen und interessante Ressourcen ausmachen. Gerade temporäre Dateien, die zudem noch gegen eine *Race Condition* verwundbar sind, können gegeben sein. Vor allem dann, wenn auf der Multiuser-Umgebung von Citrix eine Applikation zum Einsatz kommt, die eigentlich als Einzelplatz-Lösung vorgesehen ist.

Weitere Leserechte durch Directory Traversal

Beim Öffnen einer Dialogbox mittels *COMDLG32.OCX* wird ihr stets durch den Parameter InitDir das initiale Verzeichnis, welches standardmässig geladen werden soll, mitgeteilt. Hierbei handelt es sich üblicherweise um das Arbeitsverzeichnis eben jener Applikation, die von der entsprechenden Funktion Gebrauch macht. Ein alternatives Verzeichnis ist das entsprechende Heimverzeichnis des gegenwärtigen Benutzers. Die Heimverzeichnisse werden bei englischen Versionen von Microsoft Windows standardmässig unter *C:\Documents and Settings* bereitgestellt. Das Heimverzeichnis für den Benutzer mit dem Namen *mruef* lautet sodann *C:\Documents and Settings\mruef*.

Ein Angreifer könnte nun versuchen, in der Verzeichnishierarchie eine Stufe höher zu kommen. Die einfachste Möglichkeit hierzu ist der Klick auf den entsprechenden *Up one Level-Button*, der sich oben rechts findet. Jedoch kann auch über die danebenliegende Combobox ein direkter Zugriff auf höher gelegene Verzeichnis-Objekte angestrebt werden. In diesem Fall wäre dies der Zugriff auf die Heimverzeichnisse unter *C:\Documents and Settings*.

Gerade in Multiuser-Umgebungen sind Administratoren jedoch darum bemüht, durch das strikte Setzen der *NTFS*-Dateirechte (ähnlich *chmod* unter Unix/Linunx) einem unerwünschten Zugriff vorzubeugen. In diesem Fall wird beim Ansprechen eines geschützten Objekts (z.B. Verzeichnis oder Datei) eine entsprechende Fehlermeldung generiert. Ist das Auditing für das Objekt aktiviert, wird zudem ein Eintrag in der Ereignisanzeige (Event Log) erstellt.

In manchen Situationen kann es jedoch gegeben sein, dass die Administratoren aufgrund von Nachlässigkeit oder Unwissen darauf verzichten, die weiteren übergeordneten Verzeichnisse ebenfalls durch die *NTFS*-Zugriffsrechte zu schützen. Obschon beim ersten Versuch auf *C:\Documents and Settings* eine Fehlermeldung generiert und der Zugriff verweigert wurde, kann direkt versucht werden auf *C:* zu gelangen. Der Angreifer kann also darum bemüht sein, gleich mehrere Stufen höher in der Verzeichnishierarchie zu springen. Dass bestimmte Unterverzeichnisse eine Rechtelimitierung

aufgewiesen hätten, ist bei diesem nicht direkt auf diese Sub-Objekte umgesetzten Zugriffe irrelevant. Man ist quasi aus dem vorgegebenen Verzeichnis-Kontext gesprungen.

Eine Dialogbox lässt standardmässig den Wechsel in andere Verzeichnisse zu. Dies ist komfortabel durch eine Dropdown-Liste möglich, mit der man sich durch die Hierarchie der Verzeichnisse bewegen kann. Aber gerade in sicheren Citrix-Umgebungen wird diese Funktionalität gänzlich oder wenigstens teilweise abgeschaltet. Durch das Applizieren restriktiver Policies lassen sich die jeweiligen Elemente ausblenden und so den einfachen Zugriff limitieren.

Um dennoch aus dem gegebenen Verzeichniskontext ausbrechen zu können, kann man sich der klassischen *Directory Traversal*-Attacke bedienen, wie sie aus dem Web-Umfeld bestens bekannt ist. Standardmässig findet sich nämlich in der unteren Hälfte der Dialogboxen eine Textbox, die eigentlich für die manuelle Definition eines einzulesenden Dateinamens vorgesehen ist. In dieser lassen sich die oben genannten manuellen Filter anbringen. Ebenso kann damit aber auch der *Directory Tracersal*-Zugriff erfolgen.

Die klassische Variante dieser Attacke benutzt die reservierte Zeichenkette ..\, durch welche eine Stufe höher in der Verzeichnishierarchie gesprungen werden kann. Durch das Einbringen von ..\..\ ist damit das zweifache Verlassen des gegenwärtigen Verzeichnisses möglich. Mit der relativen Pfadnangabe ..\..*config.sys* lässt sich, sollte man sich im Verzeichnis *C:\Documents and Settings\maru* befinden, direkt auf die Datei *C:\config.sys* zugreifen.

Dies funktioniert natürlich auch mit komplexeren Verzeichnisangaben, bei denen Hierarchie-Stufen verlassen und wieder betreten werden können. Nehmen wir als Beispiel den Zugriff ..\..*WINDOWS**system.ini*, bei dem zuerst mit ..\..\ zwei Stufen höher gesprungen wird. Danach wird das Verzeichnis *WINDOWS* betreten und auf die Datei mit dem Namen *system.ini* zugegriffen. Ist nun dieses spezifische Zielverzeichnis und/oder die Zieldatei nicht zusätzlich geschützt, wird dieser Zugriff funktionieren.

Alternative Dialogboxen ausnutzen

In so manchem Fall kann es gegeben sein, dass die Rechteausweitung über die vordefinierten Dialogboxen nicht möglich ist. Offensichtlicherweise dann, wenn auf den Einsatz dieser verzichtet wird, was besonders bei kleineren Applikationen, die keine Interaktion mit Dateien vorsehen, der Fall ist. Oder durch das Deaktivieren der Buttons und Dropdown-Listen für das Verlassen des gegenwärtigen Verzeichnisses kann eine Rechteausweitung verhindert werden.

Kommen Dialogboxen zum Einsatz, sollte man überprüfen, ob diese sich in ihrem Verhalten unterscheiden. Denn viele Entwickler verstehen nicht, dass sowohl die Dialogbox für das Öffnen als auch jene für das Speichern einer Datei in ihren Grundzügen identisch sind. Einziger Unterschied ist, dass diese entweder mit der Methode *ShowOpen* (Öffnen) oder *ShowSave* (Speichern) angesprochen werden.

Viele Programmierer wollen nicht auf Standardkomponenten, wie die durch *COMDLG32.OCX* bereitgestellte, zurückgreifen. Stattdessen streben sie eigene Implementierungen mit ähnlicher oder erweiterter Funktionalität an. Derartige Neuentwicklungen können ein interessanter Angriffspunkt sein, wird vielleicht an dieser Stelle auf Schutzmassnahmen (z.B. *Directory Traversal*) verzichtet oder gar eine kompromittierende Rechtevererbung umsetzbar:

```
Private Sub mnuFileOpen_Click()
   'Zeige die Öffnen-Dialogbox
   cdgOpen.ShowOpen
   'Lade die angegebene Datei
   LoadFile (cdgOpen.Filename)
End Sub
Private Sub mnuFileSaveAs_Click()
   'Öffne die Save As-Dialogbox
   cdgSaveAs.ShowSave
   'Speichere die angegebene Datei
   SaveFile (cdgSaveAs.Filename)
End Sub
```

Das Thema *Directory Traversal* ist schon seit längerem fester Bestand-

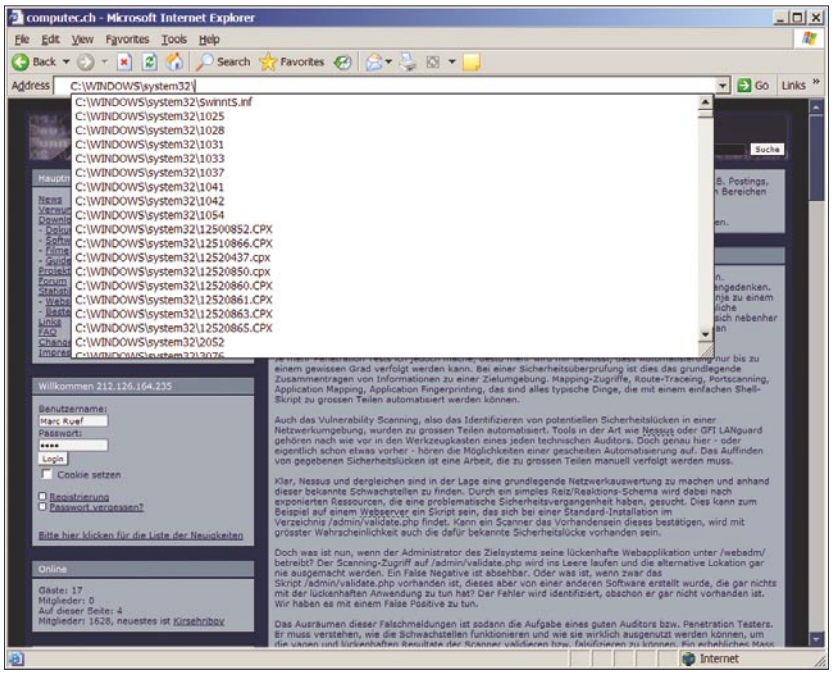


Abbildung 2. *Durch die Eingabe von Dateinamen in der Adresszeile des Internet Explorers kann der Verzeichnisinhalt angezeigt und die Existenz von Dateien ausgemacht werden. Dadurch lässt sich die Verzeichnisstruktur des Citrix-Servers ermitteln*

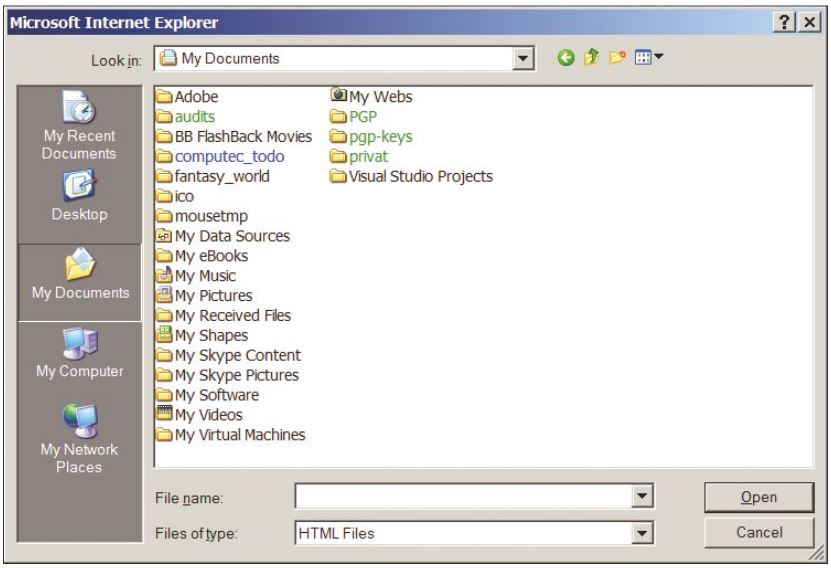


Abbildung 3. *Die von Microsoft Internet Explorer angebotene Dialogbox fokussiert sich standardmässig auf HTML-Dokumente. Durch das Ändern der Filter und das Nutzen des Kontextmenüs lassen sich jedoch erweiterte Manipulationen am Dateisystem vornehmen*

teil bei Web-Entwicklungen. Dort wird durch das Filtern entsprechender Eingaben versucht, derartige Angriffe von vornherein abzuwehren [Hoglund et al. 2004]. Auch bei lokalen Applikationen können derartige Ansätze zum Tragen kommen. Sodann verhindern sie eine Eingabe der dafür typischen ..\ Zeichenkette, die das Herausspringen aus Verzeichnissen ermöglicht. Wiederum ein einfaches Beispiel einer Öffnen-Dialogbox mit *Visual Basic 6*, bei der die Funktion `Replace()` die Schutzmassnahme übernimmt:

```
Private Sub mnuFileOpen_Click()
   'Zeige die Öffnen-Dialogbox
   cdgOpen.ShowOpen
   'Lade ohne Dir. Traversal
   LoadFile (Replace(cdgOpen.Filename,
                     "..\", "")
End Sub
```

In diesem Beispiel ist es gegeben, dass der durch die Dialogbox definierte Name nach der Zeichenkette ..\ abgesucht und diese gelöscht (eigentlich durch *nichts* ersetzt) wird. Dies führt dazu, dass ein *Directory Traversal*-Zugriff in der Form von *..\..\config.sys* zu einem schlichten *config.sys* umgewandelt wird. Ist diese Datei der Form *./config.sys* nicht vorhanden, wird der Zugriff entsprechend nicht möglich sein und eine Fehlermeldung ausgegeben.

Problem hierbei ist, dass explizit nach dem Auftreten der Zeichenkette ..\ gesucht wird. Dies macht es nämlich möglich, dass sich der *Directory Traversal*-Zugriff auch weiterhin durch das Heranziehen der Zeichenkette *../* umsetzen lässt. Diese ist in *Windows*-Umgebungen (auf der Dateisystem-Ebene) äquivalent. Desweiteren sind unter Umständen gewisse Codierungen dieser Zeichenkette möglich, um ebenfalls den Filter zu umgehen. Ein Angreifer wird also bei einer fehlerhaften Eingabeüberprüfung dank klassischer Evasion-Techniken dennoch zu seinem Ziel kommen. Eine umfassende Flankierung ist erforderlich, um keine derartigen *Directory Hopping*-Attacken zuzulassen.

Auch wenn eine annehmbare Lösung implementiert wurde, bleiben klassische Probleme der Zugriffe auf Dateiebene bestehen. Durch eine absolute Referenzierung unter Zuhilfenahme des Pfads könnten auch weiterhin Zugriffe in der Form von *Directory Traversal* angestrebt werden. Hierbei kann Beispielsweise anstatt *..\..\WINDOWS\system.ini* direkt mittels *C:\WINDOWS\system.ini* das entsprechende Objekt angesteuert werden, ohne durch die typischen Muster einer relativen Pfadangabe aufgehalten zu werden. Dies ist natürlich auch bei anderen Laufwerksbuchstaben (z.B. *D:\password.txt*) oder gar Netzwerklaufwerken (z.B. *\192.168.0.1\D\password.txt*) möglich.

Entwickler, die nicht auf vorgefertigte Bibliotheken (z.B. OCX von *Microsoft*) zurückgreifen wollen, pflegen ihre eigenen Sicherheitsmechanismen einzubringen. Ein beliebter Trick, um dort unerwünschten Eingaben zuvor zu kommen, ist das Deaktivieren gewisser Tastatur-Eingaben. Dabei werden die jeweiligen Events in `KeyDown()`, `KeyUp()` und `KeyPress()` abgefangen und anhand einer Blacklist der KeyCodes aussortiert. So kann es durchaus sein, dass in einem Feld, in dem lediglich numerische Werte erwartet werden, sämtliche Buchstaben und Sonderzeichen gar nicht erst angenommen werden. In solchen Ausnahmefällen kann manchmal mittels *Copy & Paste* oder Tastaturkombinationen trotzdem noch der unerwartete Inhalt erstellt werden. Ob und inwiefern diese Möglichkeit für einen Angreifer von Vorteil sein kann, ist danach stark von der programminternen Verarbeitung der Daten und der gegebenen Eingabevalidierung abhängig.

Die Möglichkeit einer Ausnutzung dieser Art ist jedoch nur gegeben, wenn lediglich einige wenige Schutzmassnahmen innerhalb der Zielanwendung zum Tragen kommen. Wird eine Zugriffseinschränkung auf der Betriebssystemebene (z.B. *NTFS*-Dateirechte und Group Policy) eingesetzt, lässt sich ein Angriff nicht erfolgreich umsetzen. Ein Grundsatz, der in Bezug auf die Citrix-Sicherheit von enormer Wichtigkeit ist.

Applikationen durch Dialogboxen öffnen

Dank Dialogboxen kann jedoch nicht nur auf untypische Dateitypen oder vermeintlich geschützte Bereiche zugegriffen werden. Viel mehr lassen sich damit ebenfalls Anwendungen ausführen. Dazu ist eine

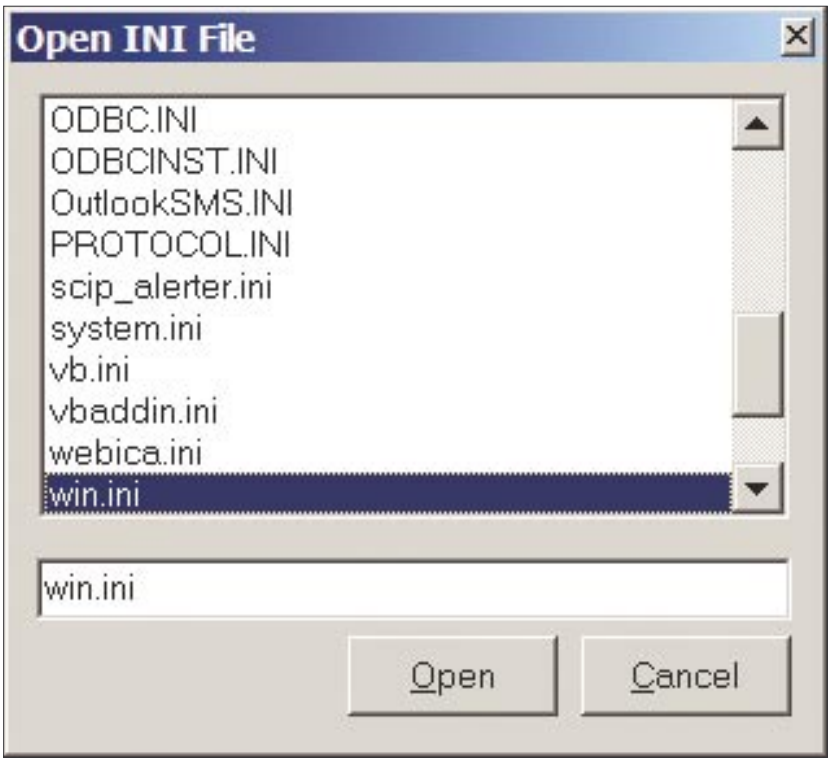


Abbildung 4. *So mancher Entwickler pflegt seine eigene Version einer Dialogbox einzusetzen. Diese können teilweise gar keine oder nur schlecht implementierte Schutzmassnahmen aufweisen. Je nachdem lassen sich im Rahmen dieser zusätzliche Manipulationen vornehmen, wie sie durch eine Standard-Dialogbox nicht hätten genutzt werden können*

Standard-Dialogbox erforderlich, auf die durch das Anpassen des Filters ausführbare Dateien angezeigt werden sollen. Dies kann beispielsweise durch das Einbringen von **.exe* oder dem absoluten * zur Anzeige sämtlicher Dateien des gegenwärtigen Verzeichnisses umgesetzt werden. Anstelle der interpretierbaren Dokumente hat man damit Zugriff auf die ausführbaren Dateien. Indem eine der ausführbaren Dateien mit der rechten Maustaste angeklickt wird, öffnet sich das Kontextmenü für diese. Dadurch lässt sich neben zusätzlichen Manipulationen (z.B. Kopieren oder Löschen) durch den Menüpunkt *Open* auch das Ausführen dieser realisieren.

Dies ist eine der Haupttechniken, durch die in einer Citrix-Umgebung erweiterte Rechte erlangt werden können. Die ausgeführte Applikation wird nämlich auf dem MetaFrame-Server gestartet und kann wie jede andere Anwendung bedient werden, auch wenn sie gar nicht freigegeben wurde.

Ein beliebtes Ziel für das Ausführen zusätzlicher Programme ist das Systemverzeichnis von *Windows* (*%windir%*), welches meistens mit *C:\WINDOWS* oder *C:\WINNT* untergebracht wird. In diesem und im Unterverzeichnis *%windir%\system32* finden sich eine Vielzahl an Systemprogrammen, mit denen sich die Auswertung und das Angreifen der Zielumgebung angehen lässt.

Oft freigegeben und nicht mit zusätzlichen *NTFS*-Rechten limitiert ist der Texteditor *notepad.exe*. Mit diesem können einfache *ASCII*-Texte verarbeitet werden, weshalb er mitunter auch von einer Vielzahl an Anwendungen mitbenutzt wird (z.B. standardmässig vom *Microsoft Internet Explorer* zur Anzeige des *HTML*-Quelltexts einer geladenen Seite).

Ebenso beliebt sind *cmd.exe* bzw. *command.com*, über die die *MS DOS*-Eingabeaufforderung erlangt werden kann. Ist dieser Schritt geglückt, können sehr einfach und effizient Manipulationen auf Dateiebene umgesetzt werden. Wie dies im Detail aussieht und mit welchen Problemen sich ein Angreifer konfrontiert sehen könnte, werden wir weiter unten besprechen.

In manchen Fällen kann es gegeben sein, dass die genutzte Dialogbox keinen direkten Zugriff auf ausführbare Dateien zulässt oder solche im Rahmen des zur Verfü-

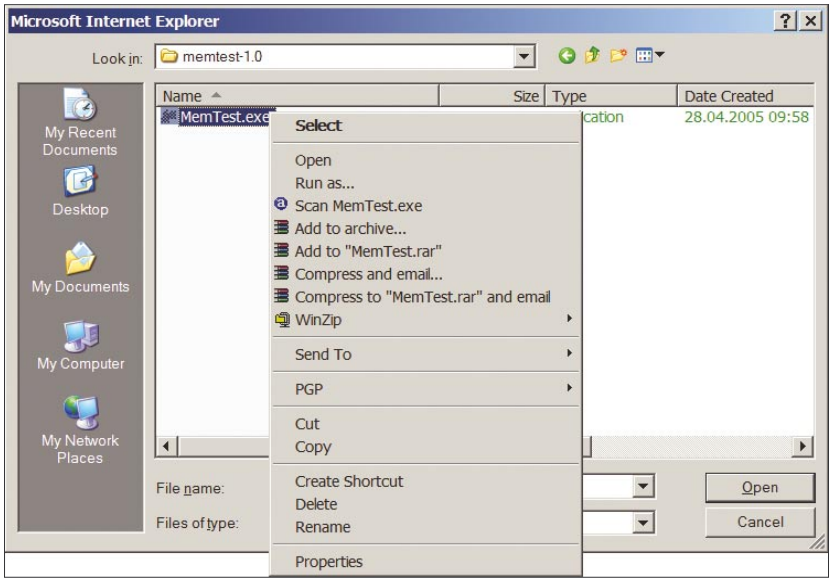


Abbildung 5. *Durch das Selektieren einer Datei und das Klicken der rechten Maustaste lässt sich das Kontextmenü im Rahmen der Dialogbox öffnen. Dadurch können Manipulationen am Objekt vorgenommen oder gar zusätzliche Applikationen gestartet werden (z.B. WinZip)*

gung gestellten Benutzerkontexts gar nicht vorhanden sind. Um dennoch weitere Anwendungen starten zu können, bedient man sich eines kleinen Tricks. Zum Beispiel das Setzen des Filters der Dialogbox von *notepad.exe* auf **.xls*, um sämtliche *Excel-Sheets* des gegenwärtigen Verzeichnisses anzuzeigen. Anstatt die neu angezeigten *XLS*-Dateien direkt mit der Dialogbox zu öffnen, wird eine solche mit der rechten Maustaste angeklickt. Im sodann geöffneten Kontextmenü sind einige Punkte für den Angriff relevant. In manchen Fällen taucht nämlich der Eintrag *Edit* auf, bei dem eine Datei sofort mit dem verknüpften Editor verändert werden kann (z.B. Bilddateien wie *BMP* und *JPEG* mit *Microsoft Paint* oder *XLS*-Sheets mit *Microsoft Excel*). Damit ist schnell eine weitere Applikation gestartet.

Einschränkungen sind dann gegeben, wenn bestimmte Dateitypen, mit denen externe Anwendungen angesteuert werden sollen, nicht vorhanden sind. Hierbei ist die Möglichkeit gegeben, dass man halt selber Dateien mit den gewünschten Erweiterungen erstellt. Von Vorteil ist hierbei die erweiterte Funktionalität, die einem die Dialogboxen zur Verfügung stellen. Diese weisen im Grunde eine ähnliche Funktionalität auf, die der Datei-Explorer selbst. Durch das Kontextmenü lassen sich Dateien kopieren, löschen und umbenennen.

Innerhalb der Dialogbox einer legitimen Applikation (z.B. *Notepad*) kann eine Datei mit einem alternativen Namen gespeichert werden. Anstelle einer herkömmlichen *TXT*-Datei kann nun beispielsweise eine *BMP*-Datei erstellt werden. Damit wird eine neue Assoziation gewährleistet und damit eine zusätzliche Applikation (in diesem Fall das simple Grafikprogramm *Paint*) startbar. So lassen sich teilweise auch Programme starten, die weder freigegeben noch anderweitig ausgewiesen, aber dennoch auf dem Citrix-Server installiert sind. Die Information dieser Zuweisungen wird simpel in der Registry in *HKEY_CLASSES_ROOT* gespeichert.

Externe Applikationen durch URIs öffnen

Eine klassische Ausweitung der Rechte innerhalb einer Citrix-Sitzung ist unter dem Miteinbeziehen von URIs gegeben. Ursprünglich führte Tim Berners-Lee den Begriff 1994 im RFC 1630 als Universal Resource Identifier ein [Barners-Lee 1994]. Erst später tauchte in offiziellen W3C-Dokumenten die Bezeichnung Uniform auf. Populärste Variante von URIs ist durch URLs gegeben, durch die sich Ressourcen innerhalb des *World Wide Webs* ansteuern lassen (z.B. *http://www.computec.ch*). Der erste Teil eines URI (vor dem Doppelpunkt), das sogenannte Schema (engl. *scheme*) gibt den Typ der Ressource an. Dabei findet innerhalb des Betriebssystems ebenfalls eine Zuweisung von Schema zu Applikation statt. Wird also eine *http*-Ressource angesprochen, wird dies standardmässig mit einem entsprechend kompatiblen Webbrowser umgesetzt. Die populärsten Schemas (engl. *scheme*), da zu grossen Teilen standardisiert, sind:

- HTTP *(Hyper Text Transfer Protocol, tcp/80 und tcp/443*); öffnet normalerweise den Standard-Webbrowser (z.B. *Microsoft Internet Explorer*) und darin die angegebene URL, Standardports;
- FTP (*File Transfer Protocol, tcp/21*); FTP-Client, falls kein solcher installiert ist, wird auf Windows-Systemen auf den Explorer zurückgegriffen;
- MAILTO (Mail); öffnet einen Mail-Client zum Versand von Email (normalerweise SMTP, tcp/25);
- NNTP (News Network Transfer Protocol, tcp/119); startet den News-Client (USENET), auf Windows-Systemen ist standardmässig keiner vorhanden;
- Gopher (tcp/70), ein schon etwas angestaubter Dienst im Stil von HTTP, als Client wird in Windows-Umgebungen ebenfalls der Microsoft Internet Explorer genutzt.

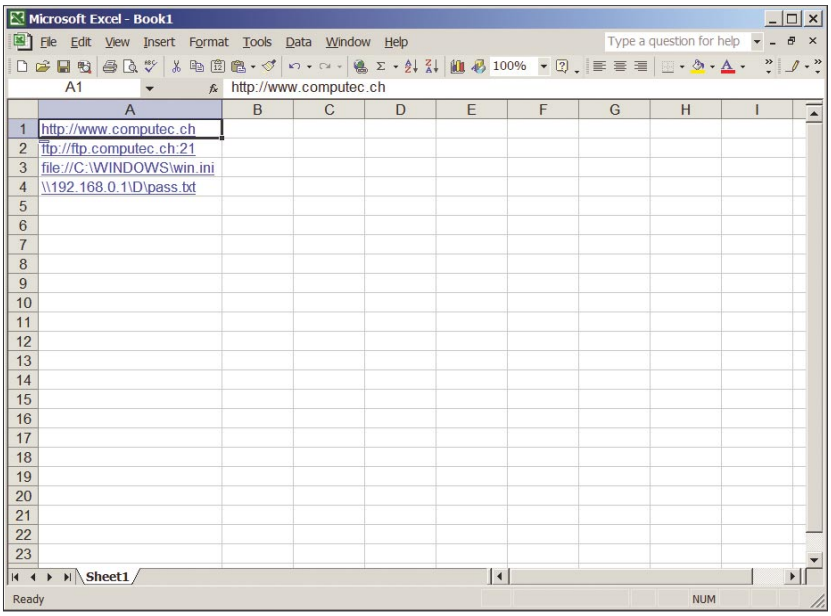


Abbildung 6. *Sehr viele professionelle Applikationen (z.B. Microsoft Excel) wandeln Eingaben, falls diese ein Schema benutzen, automatisch in Links um. Nach dieser Umwandlung können die Links angeklickt und damit eine zusätzliche Applikation, die sich für die Interpretation der Ressource verantwortlich zeichnet, gestartet werden*

Besteht auf einem System die Möglichkeit, eigene HTML-Dokumente zu erstellen und durch einen Webbrowser interpretieren zu lassen, kann man von diesen Schemas Gebrauch

machen. Sehr einfach lässt sich so etwas mit einer Datei automatisieren, in dem sich die Links für die jeweiligen Zugriffe finden. Dies kann mit **FTP-Client starten innerhalb eines HTML-Dokuments sehr einfach umgesetzt werden. Aber gerade im Zusammenhang mit dem Internet Explorer ist dies eher uninteressant, da die meisten URIs sowieso lediglich auf Komponenten dessen zurückgreifen (z.B. *http://*, *ftp://*, *gopher://*) und sich deshalb vorerst keine externen Applikationen starten lassen.

Sehr viele Anwendungen erstellen eine automatische Formatierung von URIs. Gutes Beispiel sind moderne Mailprogramme (z.B. *Microsoft Outlook*) oder erweiterte Textverarbeitungen (z.B. *Microsoft Word*). Wird in diesen eine der populären Schemas angewendet, wandelt die Applikation die Zeichenkette automatisch in einen Link um. Gibt man beispielsweise die Zeichenkette *http://www.computec.ch* ein, kann von nun an durch einen einfachen Klick auf den Link das externe Programm und darin die definierte Ressource geladen werden. Das Erstellen eines eigenen *HTML*-Dokuments mit entsprechenden Links entfällt damit.

Besonderes Interesse geniesst das in *Windows*-Umgebungen zur Verfügung gestellte Schema *file:///*. Mit dieser können lokale Dateizugriffe definiert werden. Nehmen wir einmal mehr die Tabellenkalkulation Excel als Beispiel. Geben wir dort in einer Zelle den Wert *file://C:\WINDOWS\system.ini* ein, verwandelt sich diese Zeichenkette wiederum in einen Hyperlink. Dieser verweist nun aber nicht mehr auf eine Netzwerkressource im Internet, sondern auf die lokale Datei *system.ini* im *Windows*-Verzeichnis *C:\WINDOWS*. Ein Anklicken dieses Links öffnet sodann den Explorer (*explorer.exe*), mit dem unverzüglich erweiterte Rechte genossen werden können. Eine komfortable Einsicht und Manipulation des Systems wird möglich. Um den Datei-Explorer samt Anzeige eines Verzeichnis-Inhalts zu erzwingen, kann einfach ein file-Zugriff auf einen Verzeichnis-Namen, beispielsweise *file://C:\WINDOWS* für das Verzeichnis *C:\WINDOWS* initiiert werden.

Durch die Referenzierung mittels absoluten Pfaden lassen sich auch direkt ausführbare Dateien starten. Um das Notepad zu starten, kann simpel auf *file://C:\WINDOWS\system32\notepad.exe* verwiesen bzw. zugegriffen werden, sofern es sich um eine Standard-Installation jüngeren Datums handelt.

Eine jede Applikation sieht sich theoretisch in der Lage, eigene URIs zu registrieren, sofern natürlich beim Installations-Prozess die dazu erforderlichen Rechte vorhanden waren. Beispielsweise registriert sich die populäre *Voice-over-IP* Anwendung Skype (*http://www.skype.com*) die URIs *skype://* und *callto://*. Eine Liste der gängigsten URIs kann nützlich sein, um die potentiell dadurch aufrufbaren Anwendungen durchprobieren zu können.

Zugriffsschutz auf cmd.exe umgehen

In vielen Umgebungen verhindern es Administratoren, und dies aus guten Gründen, dass die Benutzer direkten Zugriff auf eine Kommandozeile erhalten [*Microsoft 2004*]. Innerhalb von Windows-Umgebungen wird dabei die Möglichkeit des Aufrufs von *cmd.exe*, meistens durch eine Limitierung auf dem Betriebssystem (Group Policy oder *NTFS*-Rechte), eingeschränkt. Wird über die Kommandozeile zur interaktiven Verarbeitung initiiert, generiert dies eine entsprechende Fehlermeldung. Der Benutzer wird sodann auf die Restriktion hingewiesen und muss mit einem Abbruch des gewünschten Zugriffs vorlieb nehmen.

Dabei wird gerne übersehen, dass die Datei *cmd.exe* historisch gewachsen ist. Eingeführt wurde sie innerhalb von *Microsoft Windows NT* und übernommen in allen professionellen Folgeversionen von *Windows* (*Windows 2000*, *XP* und *Server 2003*). Bei den älteren und privaten *Windows*-Versionen ist hingegen *command.com* für das Bereitstellen des Kommandozeileninterpreters zuständig. Dieses Relikt vergangener *MS DOS*-Tage wird jedoch zwecks Rückwärtskompatibilität ebenfalls auf neueren *Windows*-Systemen angeboten. Interessant ist dies deshalb, dass die über *appsec* umgesetzten Restriktionen, die für *cmd.exe* greifen, nicht ebenfalls automatisch auf *command.com* appliziert werden. Ebenso übersehen viele Administratoren auch, dass sich dadurch eine Alternative für Zugriffe bietet, die ebenso abgedichtet werden muss.

Das Ausweichen auf das klassische *command.com* bringt dem Angreifer lediglich marginale Nachteile. So wird dabei auf eine History-Funktion für die Eingaben, wie man es von den modernen *Unix-Shells* her kennt, verzichtet. Ebenso werden die langen Datei- und Verzeichnisnamen, wie sie mit *VFAT/FAT32* und *NTFS* eingeführt wurden, in der Kurzform (8.3-Schreibweise) dargestellt, was besonders bei komplexen Verzeichnisstrukturen relativ unkomfortabel sein kann. Zudem muss mit einer schlechteren Performance der 16-bit Applikation gerechnet werden [*Microsoft 2005*]. Dies sind aber sowieso Dinge, die für einen Angreifer zum gegebenen Zeitpunkt sowieso zweitrangig sind.

In manchen Fällen sind die Zugriffslimitierungen für die Kommandozeilen *cmd.exe* und *command.com* explizit auf ihre absoluten Pfade angewandt worden. Dies bedeutet, dass auf dem Windows-System der absolute Pfad, zum Beispiel *C:\WINDOWS\system32\cmd.exe*, vermerkt wird. Dies eröffnet die Möglichkeiten, eine Kopie der ursprünglichen Datei anzulegen. Speichert man diese zum Beispiel neu unter *C:\temp\shell.exe* ab, kann die Blacklist-Funktionalität nicht mehr greifen. Sodann sieht man sich in der Lage, eine zuvor gesperrte Datei auszuführen.

Aufgrund strenger Limitierungen im Kopieren und Verschieben von Dateien kann es sehr schwierig sein, eine Adaption einer solchen anzulegen. In derartigen Situationen

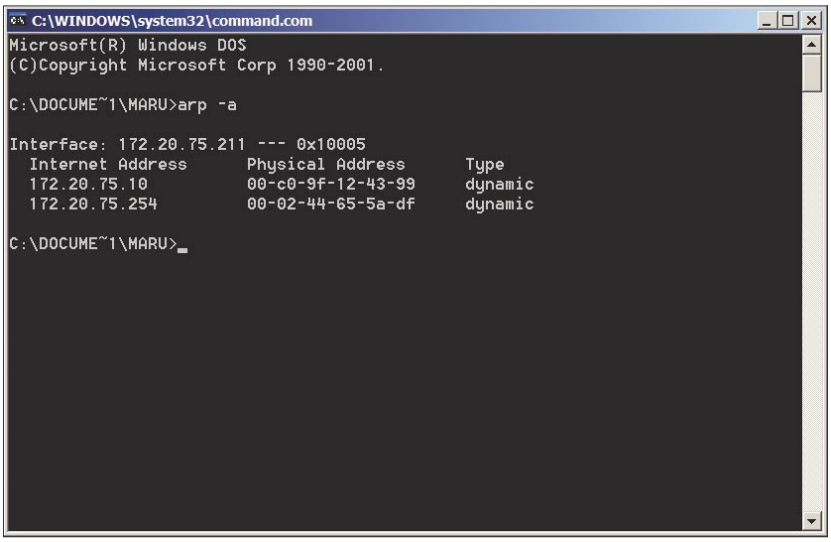


Abbildung 7. *Obschon in mancherlei Umgebungen der Zugriff auf cmd.exe limitiert wird, übersehen Administratoren gerne das Potential von command.com. Durch dieses Relikt aus MS DOS-Zeiten können ebenso komfortabel wie effizient Manipulationen auf dem Dateisystem vorgenommen werden*

ist es nützlich, wenn man sich die gewünschten Dateien anderweitig auf den Server laden kann. Oftmals ist dies durch den *Microsoft Internet Explorer* möglich, durch den die entsprechenden Daten auf die Festplatte des Citrix-Systems gezogen werden können. Von Vorteil in solchen Situationen ist es, wenn entsprechend irgendwo im Internet ein Server steht, der die jeweiligen Versionen von *cmd.exe* oder *explorer.exe* zur Verfügung stellt. Eine mit einem spezifischen *Windows*-System mitgelieferte Datei kann nämlich nicht so ohne weiteres in einer anderen *Windows*-Umgebung – sofern die gleiche Version zum Einsatz kommt - genutzt werden.

Gleiches ist natürlich auch dann gegeben, wenn *Whitelists* herangezogen werden, bei denen einzelne Programme freigeschaltet sind. Ausnahmen werden nämlich gerne für *notepad.exe* gemacht, das sich auf den meisten interaktiven Systemen früher oder später als nützlich erweist. Ein Angreifer könnte in diesem Fall eine beliebige *EXE*-Datei auf den Citrix-Server laden (z.B. einen Exploit oder ein Debugging-Utility), diese dort als *notepad.exe* abspeichern und damit ausführen lassen.

Vererbung von Rechten nutzen

Viele Entwickler und Administratoren von Software-Lösungen sind darum bemüht, dass die gewährleisteten Applikationen ausschliesslich auf jene Ressourcen zugreifen können, die zwingend erforderlich sind. Dieser Ansatz verhindert grundsätzlich, dass unnötigerweise Rechte zugesprochen werden, die einen Missbrauch zu fördern in der Lage sind.

Dennoch kann es einmal vorkommen, dass diesem förderns-werten Bestreben nicht richtig oder umfassend nachgekommen wird. Vor allem in komplexen Produkten ist die Chance gross, dass bei der Umsetzung eines restriktiven Verhaltens Fehler unterlaufen. Manchmal kann es aber auch gegeben sein, dass zur Gewährleistung bestimmter Funktionalitäten kurzweilig erweiterte Rechte eingebracht werden sollen. Vor allem, wenn externe Komponenten wie Bibliotheken eingebunden werden, kann es zu designtechnischen Nachlässigkeiten und Fehlern kommen.

Besonders davon betroffen sind systemnahe Anwendungen. Dazu zählen vor allem Server-Dienste, die sich oftmals tief im System einnisten und zur Erledigung ihrer Arbeiten die zusätzlichen Rechte (oftmals *Administrator/SYSTEM*) erfordern. Hat ein Angreifer einmal eine solche Applikation initiiert und erlaubt diese zusätzliche Zugriffe, ist es nur noch eine Frage der Zeit, bis der Host unter totaler Kontrolle ist. [*Windows* About 2006]

Aber auch die hauseigenen Anwendungen von *Microsoft* sind in diesem Belang von des Angreifers Interesse. Dies liegt vor allem daran, weil die vermeintlich anwenderfreundlichen Applikationen eine Vielzahl an Hintergrund-Aufgaben erledigen, die wiederum zusätzliche Rechte erfordern. Dies reicht vom Mitlesen der Inhalte anderer Fenster bis hin zur dynamischen Manipulation sensitiver Registry-Einträge. Gerade wenn sich ein Angreifer den *Microsoft Internet Explorer*, *Word* oder *Excel* einverleiben konnte, ist es mittelfristig um das Citrix-System geschehen.

Eine Vererbung von Rechten durch das Anwenden verschiedener Angriffsformen kann zur Kompromittierung eines Systems führen. Als Beispiel sei in einer Citrix-Umgebung ausschliesslich der Webbrowser *Microsoft Internet Explorer* freigegeben. Durch die Speichern-Dialogbox dessen und dem Nutzen eines *Directory Traversal* - Zugriffs kann auf die Datei *command.com* zugegriffen werden. Ein Ausführen dieser ist jedoch aufgrund der gegebenen Policy nicht möglich. Durch das Kopieren und Umbenennen der Datei innerhalb der genutzten Dialogbox kann es jedoch möglich werden, die *MS DOS*-Eingabeaufforderung in gewünschter Weise nutzen zu können.

Besonderes Interesse vererbter Rechte geniessen vor allem Hilfe-Funktionen, die bei der Rechtelimitierung gerne durch Administratoren vergessen oder übergangen werden. Unter Umständen werden eigene Hilfe-Utilities eingesetzt und diese wiederum erlauben das manuelle Öffnen von Dateien. Oftmals muss man Umwege über verschiedene Features machen, bis man eine Hilfe-Funktion findet, die auch wirklich von Nutzen ist – Dann springt man

halt von Hilfe zu Hilfe. Vor allem die Hilfe-Anzeigen von Drucker-Einstellungen sind dafür bekannt, dass sie auf proprietäre und schlecht geschriebene Hilfe-Utilities zurückgreifen. Aber selbst die *Microsoft Windows Help 5* erlaubt so manche Rechteausweitung.

Erweiterte Rechte durch Tastenkombinationen

Wie zuvor erwähnt, sind Administratoren darum bemüht, dass in einer Citrix-Umgebung möglichst nur jene Ressourcen zugänglich sind, die für den Benutzer von Wichtigkeit sind, damit dieser die ihm aufgetragenen Arbeiten erledigen kann. Dies führt dazu, dass oftmals lediglich einige wenige Applikationen, wie zum Beispiel *Microsoft Excel* oder eine Anwendung zur Buchhaltung, freigegeben sind. Das größte Ziel für einen Angreifer in einer solchen Umgebung ist und bleibt dabei der gesamte Desktop-Zugriff. Hat er diesen erreicht, steht ihm die komplette interaktive Umgebung, wie man sie von einer lokalen *Windows*-Sitzung her kennt, zur Verfügung.

Eine solche Rechteausweitung kann über verschiedene Wege angegangen werden. Die populärsten Methoden sind mit Sicherheit die Ausnutzung von Dialogboxen und das Einbringen von automatischen Links (URIs). Dadurch bleibt es möglich, dank der Interaktion mit einer verwundbaren Applikation weitere Ressourcen für den eigenen Vorteil einzunehmen.

In den letzten Jahren war im Rahmen der Software-Ergonomie die Tendenz zur Grafik spürbar. Durch intuitive Benutzeroberflächen sollte die Handhabung entsprechender Betriebssysteme und Applikationen möglichst einfach und effizient gestaltet werden können. Sowohl *Microsoft Windows* als auch *Apple MacOS X* haben hier ein neues Zeitalter eingeläutet und damit den Computer auch für die breite Masse attraktiv gemacht.

Einer der Trends grafischer Oberflächen ist, dass die Eingabe durch den Benutzer nach Möglichkeiten durch die Maus erfolgt. Komplexe Kommandoeingaben, wie sie gerne in den Shells von unixoiden Betriebssystemen ihre Verwendung finden, rücken dabei immer mehr in den Hintergrund. Und muss einmal für eine Interaktion die Tastatur herangezogen werden, hat dies besonders einfach zu geschehen. Viele Betriebssysteme und Applikationen arbeiten sodann mit Tastaturkombinationen. Durch das Drücken spezieller Tasten ist es möglich, die genutzte Anwendung zu einen bestimmten Verhalten zu bewegen. Bestes Beispiel ist durch die Taste *F1* gegeben, die in den meisten Anwendungen die Hilfe startet. Oftmals erlaubt der Start dieser zusätzlichen Anwendung das Erweitern der Rechte enorm. Vor allem dann, wenn sich damit nach Belieben andere Dateien (z.B. *HTML*-Dokumente), die sich eventuell gar noch vorgängig selber erstellen lassen, öffnen lassen.

Eine weitere Tasten-Kombination, die in dieser Hinsicht immer wieder für Angreifer von Interesse ist, ist im Drücken der *Windows*-Taste und der Taste *E* gegeben. Dies öffnet unverzüglich und praktisch aus jeder Umgebung heraus, den *Explorer*, mit dem sich fortan sehr einfach Dateizugriffe umsetzen lassen. Ebenfalls einen Schritt weiter geht die Tasten-Kombination *Shift+Ctrl+Esc*, mit der sich nach Belieben eine Instanz des Taskmanagers ausführen lässt. Aus diesem Heraus lassen sich wiederum ganz einfach Prozesse beeinflussen oder neue mittels File/New Task starten.

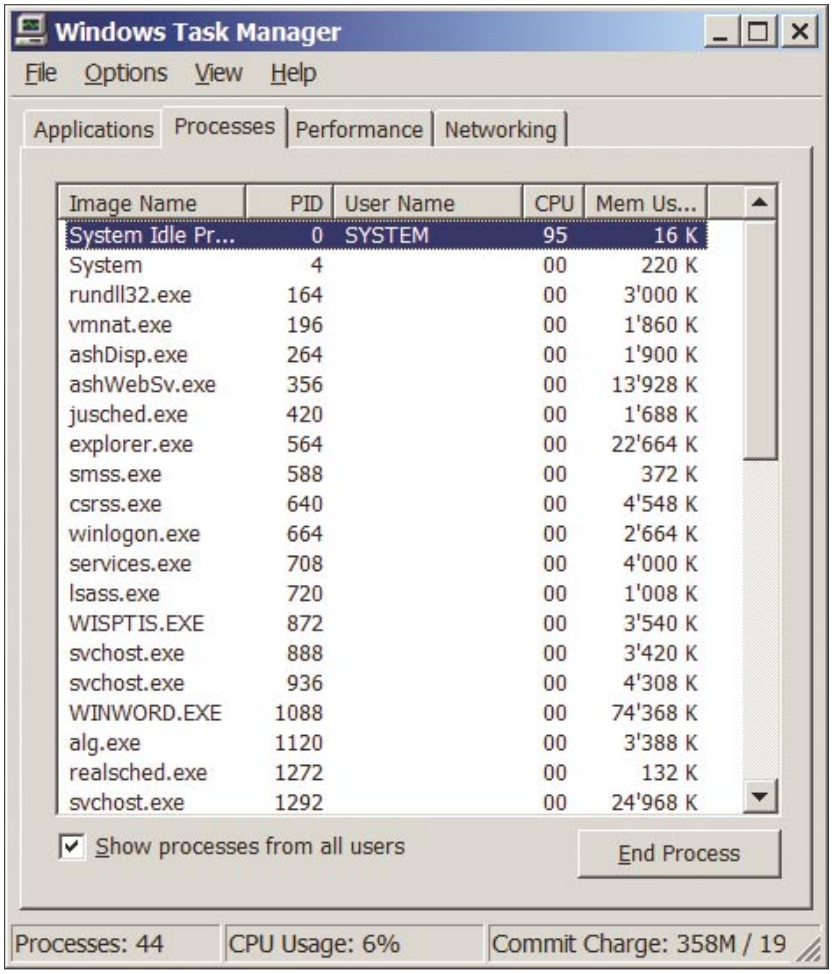


Abbildung 8. *Mit Tastenkombinationen lassen sich gut und gerne Zusatzkomponenten oder weitere Applikationen starten. Ein Angreifer könnte darum bemüht sein, den Taskmanager aufzurufen und aus diesem über File/New Task eine neue Anwendung zu starten. Dadurch lassen sich weitere Rechte erlangen*

Erweiterte Rechte durch Skripting

In manchen Fällen kann es gegeben sein, dass einem Citrix-Benutzer direkt oder indirekt die Möglichkeiten von Programm- und Skript-Entwicklungen gegeben sind. Offensichtliches Beispiel ist eine Umgebung, in der die Werkzeuge zur *.NET*-Entwicklung bereitgestellt werden. Durch Editoren und Compiler können eigene Applikationen geschrieben und auf dem Citrix-System ausgeführt werden. Es scheint offensichtlich, dass sich damit und entsprechender Fleissarbeit eigene Anwendungen erstellen lassen, die damit auch zusätzliche Rechte zu versprechen in der Lage sind.

In diesem Zusammenhang wird aber immer wieder gerne vergessen, dass auch eine Vielzahl an harmlosen Applikationen mit einer derartigen Automatisierung-Funktionalität aufwarten können. An erster Stelle seien hierbei die kommerziellen Lösungen aus der Office-Reihe von *Microsoft* zu nennen. In den Produkten wie *Microsoft Word* oder Excel können *VBA*-Skripte (*Visual Basic for Applications*) entwickelt und ausgeführt werden. Damit ist quasi eine abgespeckte Version einer Entwicklungsumgebung vorhanden. Dass sich mit derartigen Features Angriffe umsetzen lassen, haben Wurm-Epidemien wie *ILOVEYOU* und *Melissa* medienwirksam und damit eindrucksvoll bewiesen. Unter Umständen ist die Rechtevergabe für derartige Skripte viel laxer ausgelegt, weder für die manuellen Zugriffe des Angreifers selbst.

Noch eine Stufe primitiver, aber in einigen Situationen nicht minder erfolgreich, kann das Schreiben von *Batch*-Skripten sein. Durch einen herkömmlichen Editor lassen sich kleine Programme schreiben, mit denen Auswertungen oder gar als Angriff klassifizierbare Zugriffe automatisieren lassen.

Auch wenn sich nicht direkt erweiterte Rechte durch ein selber geschriebenes Skript einholen lassen, können damit doch eine Vielzahl an stupiden Angriffsversuchen automatisiert und damit erleichtert werden. *Bruteforce*-Attacken auf Anwendungen oder lokale Dienste sind genauso denkbar, wie das automatisierte Auslesen von *NetBIOS*-Namenstabellen im lokalen Netzwerk. Im Folgenden sollen die Möglichkeiten derartiger Zugriffe illustriert werden.

Systemnahe Exploits

Wir haben bisher sehr viele manuelle Überprüfungen, die sich grundsätzliche, konzeptionelle oder administrative Schwachstellen einer Umgebung zunutze machen, betrachtet. Dies sind mit Sicherheit jene Schwachstellen, die explizit in Multiuser-Umgebungen – wie Citrix nun mal auch eine ist – angetroffen werden. Alternative Herangehensweisen sind im Heranziehen spezifischer Exploits zu finden. Dabei versucht ein Angreifer, eine spezifische Schwachstelle in einer Anwendung auszunutzen, um erweiterte Rechte zu erlangen. Wir kennen dies vor allem in Netzwerkumgebungen, in denen ein Angreifer beispielsweise eine Pufferüberlauf-Schwachstelle in einem Apache-Webserver ausnutzt, um administrative Privilegien zu erschleichen. Das gleiche Prinzip kann auch auf Hostsicherheit angewandt werden.

Es ist wichtig zu verstehen, dass bei der Arbeit mit Citrix direkter Einfluss auf den Server ausgeübt werden kann. Dabei ist das Server-System natürlich darum bemüht, den Benutzer möglichst in seinem dedizierten Kontext arbeiten zu lassen und ihm dadurch Zugriffe auf systemkritische Bereiche verwehren zu können. Dennoch ist mit der Zugriffsmöglichkeit, die durch Citrix gegeben ist, ein sehr nahes Arbeiten am System möglich.

Schutzmassnahmen auf der Netzwerkebene greifen in diesem Falle nicht mehr. Eine dedizierte Perimeter-Firewall schützt primär vor Benutzern, denen es nicht gestattet ist, in das sichere Netz einzudringen. Ein legitimer Citrix-Benutzer muss und wird jedoch von einem solchen Zugriff nicht abgehalten, denn ansonsten könnte er seiner Tätigkeit nicht nachkommen. Die Firewall (z.B. ein Paketfilter) ist also in diesem Zusammenhang eher nutzlos, agiert sie doch in erster Linie als dummes *Routing*-Element, das schlichtweg einen Verbindungsknoten zwischen zwei Netzen darstellt.

Von Interesse für einen Angreifer ab dem legitimen Zugriff auf dem Citrix-System sind jene Schwachstellen, die erweiterte Rechte versprechen und sich primär lokal ausnutzen lassen. Von größtem Interesse sind dabei Applikationen, die mit erweiterten Rechten (z.B. als *SYSTEM*) betrieben werden. Da bei einem erfolgreichen Übergriff die Rechte der missbrauchten Applikation vererbt werden, gestaltet sich derartige Zugriffe von höchstem Masse lukrativ. Gerade weil die Betriebssystemreihe von *Microsoft* so populär ist, werden auch besonders viele Schwächen im Kern gefunden. Dabei liegt der Fokus der Hacker-Szene jedoch eher im Netzwerk-Bereich, da derartige Fehler eine netzübergreifende Übernahme eines Hosts versprechen. Wohingegen lokale Schwachstellen eher in Ausnahmefällen wirklich von Bedeutung sind (z.B. wenn der Host schon kontrolliert wird oder bei legitimen Remote-Zugriffen).

Gerade Antiviren-Lösungen sind besonders anfällig, wenn es um Attacken auf das Dateisystem geht. Da derartige Produkte in der Regel die Zugriffe auf Dateien überwachen, können manipulative Bestrebungen direkten Gewinn ermöglichen. Ein einfaches Beispiel ist das Scannen eines Verzeichnisses auf mögliche Infektion durch Computerviren. Bietet die eingesetzte Antiviren-Lösung nun die Möglichkeit an, dass man sich mit einem *Explorer* durch das Dateisystem navigieren kann, würde ein Angreifer als erstes überprüfen, ob man mit diesen *Explorer* nun auch explizite Dateimanipulationen (z.B. Öffnen, Kopieren, Verschieben, Löschen) anstellen kann. Falls dem so ist, dann könnte es durchaus sein, dass er erweiterte Rechte der Applikation erbt, die beispielsweise Zugriffe auf Bereiche und Dateien ermöglicht, für die er sonst gar keine Privilegien gehabt hätte. In einem

solchen Fall liesse sich sehr schnell das System durch gezielte Eingriffe unter Kontrolle bringen.

Gerade weil die netzwerkbasierten Schutzmechanismen nicht mehr greifen können, bieten sich die sogenannten Loopback-Attacken an. Hierbei führt ein Angreifer einen Netzwerkzugriff auf das lokale System durch. Damit können wir Netzwerkdienste, die sich sonst ganz regulär in einem LAN ansprechen liessen, auch vom eigenen System aus nutzen. Ist einer der angebotenen Dienste gegen eine Attacke verwundbar (z.B. *Directory Traversal* oder Pufferüberlauf), kann diese Schwachstelle genauso ausgenutzt werden, um erweiterte Rechte zu erlangen. Halt einfach mit dem Unterschied, dass hierbei nicht wirklich ein Remote-System, sondern der lokale Host angegriffen wird.

Hopping Angriffe

Durch einen erlaubten Zugriff auf ein Citrix-System wird damit ein direktes Arbeiten im Zielsegment möglich. Da sich das Arbeiten dabei so gestaltet, als sässe man direkt vor dem System selbst, sind ebensolche Angriffe innerhalb des Netzwerks denkbar. Wie zuvor beschrieben, hält die Perimeter-Firewall den Zugriff für erlaubt und leitet die Anfragen entsprechend weiter. Sie kann also weitere Attacken innerhalb des Zielsegments nicht mehr verhindern.

Dies macht es möglich, dass ein einmal kompromittiertes – auch wenn nur teilweise – Citrix-System als Hop für weitere Angriffe missbraucht werden kann. Offensichtlich ist es, dass innerhalb der gleichen Zone nach weiteren Angriffszielen gesucht und sich auf diese konzentriert wird. Die klassischen Auswertungs- und Angriffstechniken kommen hierbei zum Tragen (z.B. Ping und nbtstat-Auswertungen). Ausgeführt werden sie auf dem Citrix-System selbst.

Besonders aufgrund ihrer Einfachheit beliebt sind Zugriffe auf NetBIOS-Freigaben. Durch das Nutzen von *\192.168.0.10* im Rahmen einer Instanz eines Explorers auf dem Citrix-System kann versucht werden, eine Verbindung zum System mit der IP-Adresse 192.168.0.10 herzustellen. Sind bei diesem entsprechen-de NetBIOS-Freigaben vorhanden, kann auf diese zugegriffen werden. Unter Umständen, besonders bei fehlerhaft vergebenen Rechten, kann so in den Besitz sensitiver Informationen gekommen werden. Komplett freigegebene Laufwerke sind in größeren Unternehmungen, bei denen auf die Sicherheit der Perimeter-Firewall vertraut wird, keine Sicherheit. Dass man damit einem *internen* Angreifer schutzlos ausgeliefert ist, ist dabei nicht bewusst.

Ist der Citrix-Server in einer DMZ positioniert, kann darauf spekuliert werden, dass das eingesetzte Firewall-Element den einen oder anderen Zugriff in weitere Zonen zulässt. Gerade weil die Interaktion mit einem Citrix-Server, besonders wenn er in eine komplexe Windows-Umgebung eingebunden ist, relativ unüberschaubar ist, pflegen viele Administratoren mal eben sämtliche Kommunikationen zuzulassen. Dass sich sodann eine Vielzahl an Zugriffen auf strukturell wichtige Systeme umsetzen lassen (z.B. Dateiserver oder Nameserver), ist nichts ungewöhnliches.

Gerade die Kommunikationen über die Ports 135, 137 bis 139 sowie 445 werden oftmals in Windows-Umgebungen vorausgesetzt und sind deshalb freigegeben. Aber auch genau über diese Ports liessen sich in der Vergangenheit eine Vielzahl der wirklich kritischen und schlussendlich besonders einfachen Angriffe auf Windows-Systeme umsetzen. Eine schlechte und fehlerhafte Segmentierung eines Netzwerks kann dann dazu führen, dass von einem Citrix-System aus das gesamte Netzwerk übernommen werden kann.

Erweiterte Rechte: Was nun?

Die in dieser Publikation besprochene Situation nimmt einen legitimen Citrix-Zugang, bei dem nur einige wenige Applikationen freigegeben sind, als Ausgangslage. Das Ziel des Angreifers, der sich in diesem Moment als legitimer Benutzer sieht, ist das Ausweiten seiner Rechte. Durch das Aufstarten zusätzlicher Anwendungen kann mehr und mehr die Kontrolle über den Citrix-Server erlangt werden. Das höchste Ziel dabei ist ein priviligierter Zugang, bei dem administrative Rechte gegeben sind. Ist das System unter Kontrolle, lassen sich weitere Angriffe in der Zielumgebung, entweder in der gleichen Zone oder auf Systeme in anderen Netzwerksegmenten, anstreben. [Ruef 2001]

Die Rechteausweitung geniesst in der Regel sehr hohe Priorität. Erst wenn das gesamte System unter Kontrolle ist, sind die meisten Angreifer zufrieden. Dies bedeutet, dass sie sich administrative Privilegien zuweisen können müssen. Sobald dieses Ziel erreicht ist, können die weiteren Phasen eines Angriffs initiiert werden. Diese sind im Allgemeinen die Folgenden [Ruef 1999, McClure et al. 2005]:

- Daten entwenden;
- Ressourcen nutzen;
- Hintertüren einbringen;
- Spuren verwischen.

Schutzmassnahmen

Das Absichern eines Citrix-Systems ist sehr schwierig [Dannbacher und Kienle 2002]. Das Hauptproblem besteht darin, dass dem Windows-Betriebssystem eine Multiuser-Funktionalität aufgezwungen wird, die es in Bezug auf die Sicherheit nicht adequat gewährleisten kann. Microsoft Windows wurde als Einzelplatzsystem entwickelt und in den letzten Jahren lediglich mit Funktionen erweitert. Die Architektur der Plattform ist, aufgrund der gewünschten Rückwärtskompatibelität, zu großen Teilen die gleiche.

In jenem Moment, in dem einem Benutzer Zugriff auf ein Citrix-System gewährt wird, kann er grundsätzlich mit diesem interagieren, als sässe er direkt davor. Dies bedeutet, dass er in erster Linie lokale Angriffe anstreben kann. Die Absicherung eines Systems vor lo-

kalen Angriffen ist hingegen enorm kompliziert, da im Gegenzug zu netzwerkbasierten Lösungen nicht ausschliesslich ein zentralisierter Kommunikationskanal überwacht und geschützt werden muss (z.B. bei einem Webserver).

Eine sichere Citrix-Installation erfordert grundsätzlich, dass das genutzte Betriebssystem ebenfalls möglichst sicher ist. Dies bedeutet, dass das genutzte Windows so abgespeckt wie möglich installiert werden sollte. Unnötige Software-Pakete und Dienste stören nur. Sie können neben Performance-Einbussen die Sicherheit der gesamten Umgebung gefährden. Entsprechend installiert man sie am besten gar nicht, deinstalliert sie nachträglich oder deaktiviert sie wenigstens. Das Härten und Patchen eines Citrix-Servers ist unerlässlich, um sich gegen altbekannte und publik gewordene Schwachstellen zu schützen. Dadurch lässt sich das Zeitfenster für erfolgreiche Angriffe, die sich eventuell durch einfache Exploits aus dem Internet automatisieren lassen, minimieren. Ein solides *Hardening*-Konzept sowie *Change-/Release-Management* hilft dabei, die technischen Anpassung prozessgesteuert umsetzbar zu machen.

Die Schutzmassnahmen bei einem lokalen Zugriff, wie er bei Citrix gegeben ist, müssen im Gegensatz zu reinen Netzwerk-Diensten erweitert werden. Dies bedeutet, dass die Absicherung des Betriebssystems auf der Dateiebene mindestens so wichtig wird. Das Vergeben strikter *NTFS*-Rechte und die Restriktion von Registry-Zugriffen wird erforderlich, um die Rechteausweitung nach Möglichkeiten zu verhindern. Dies lässt sich leider nur, will man es richtig machen, in akribischer Feinarbeit realisieren. Der Aufwand dessen bei der Konzeption und der technischen Umsetzung sind nicht zu unterschätzen.

Zusätzliche hostbasierte Sicherheits-Installationen sind ebenfalls empfehlenswert [Ruef 2002]. Durch Personal Firewalls oder spezifische *IPsec*-Filter können unerwünschte Zugriffsversuche – auch über die *Loopback*-Schnittstelle – verhindert werden. Über das lokale Netzwerk initiierte Attacken lassen sich so abwehren (z.B. mit *telnet localhost 25*). Können die Anwender eigene Dateien herunterladen oder Programme installieren, lohnt sich die Installation einer Antiviren-Lösung. Diese sollte ebenfalls auf dem neuesten Stand gehalten werden und im Mindesten die ausführbaren Dateien nach korruptem Programmcode untersuchen.

Konnte ein annehmbares Mass an Grundschutz auf der Betriebssystem-Ebene erreicht werden, kann man sich um die Sicherheit von Citrix an sich kümmern. Es sollen stets so wenige Applikationen wie möglich freigegeben werden, um den Handlungsspielraum für Auswertungen und die Angriffsfläche für spezifische Attacken so klein wie möglich zu halten.

Gerade weil ein sehr hohes Mass an Schutzfunktionen gewährleistet werden muss, zu deren Umsetzung viel Arbeit betrieben werden muss, lohnt sich eine umfassende Risikoanalyse. So ist unter Umständen gegeben, dass die Integrität des Citrix-Systems nicht über dem

Im Internet

- *http://www.computec.ch/download.php?view.344* - Dannbacher, André, Kienle, Fabian, Oktober 2002, Leitfaden für sichere Citrix-Systeme;
- *http://www.amazon.de/exec/obidos/ASIN/0201786958/* - Hoglund, Greg, McGraw, Gary, Februar 2004, Exploiting Software – How to break Code, Addison-Wesley Professional, ISBN 0201786958,
- *http://www.amazon.de/exec/obidos/ASIN/3826681673/* - McClure, Stuart, Scambray, Joel, Kurtz, Georg, November 2005, Das Anti-Hacker-Buch, Vmi Buch, ISBN 3826681673, englische Original-Ausgabe erschienen mit dem Titel *Hacking Exposed*
- *http://support.microsoft.com/kb/184006/de* - Microsoft, 20. Januar 2005, Für das FAT32-Dateisystem geltende Beschränkungen, Microsoft.de;
- *http://support.microsoft.com/kb/320181/de* - Microsoft, 17. März 2004, SO WIRD'S GEMACHT: Verwenden des Tools „Application Security", um den Zugriff auf Programme in den Terminaldiensten von Windows 2000 einzuschränken, Microsoft;
- *http://www.computec.ch/download.php?view.283* - Ruef, Marc, 1999, Die Sicherheit von Windows, *astalavista.com* und *computec.ch*;
- *http://www.computec.ch/download.php?view.110* - Ruef, Marc, 22. Dezember 2001, Die psychosozialen Aspekte der Computerkriminalität, *computec.ch*;
- *http://www.computec.ch/download.php?view.302* - Ruef, Marc, 2002, Intrusion Prevention – Neue Ansätze der Computersicherheit, Computer Professional, Ausgabe 4-2002, Seiten 10-14;
- *http://www.computec.ch/download.php?view.528* - Ruef, Marc, 2003, Auditing mit Linux - Entdecken Sie die Schwachstellen Ihres Systems, *computec.ch;*
- *http://www.computec.ch/download.php?view.481* - Ruef, Marc, Februar 2004, Lehrgang Computersicherheit, Universität Luzern, Master of Advanced Studies eLearning und Wissensmanagement;
- *http://www.computec.ch/download.php?view.747* - Ruef, Marc 8. Februar 2006, Citrix under Attack, scip monthly Security Summary, Ausgabe 19. Februar 2006, *computec.ch;*
- *http://www.amazon.de/exec/obidos/ASIN/381582284X/* - Ruef, Marc, Rogge, Marko, Velten, Uwe, Gieseke, Wolfram, November 2002, Hacking Intern - Angriffe, Strategien, Abwehr, Data Becker, Düsseldorf, ISBN 381582284X;
- *http://www.scip.ch/cgi-bin/smss/showadvf.pl?id=1510* - scip AG, 1. Juni 2005, ZoneLabs ZoneAlarm 5.x Vet engine bis 11.9.1 Vet Antivirus Engine VetE.dll OLE-Stream Pufferüberlauf, scip AG;
- *http://windows.about.com/library/tips/bltip696.htm* - Windows About, 6. Februar 2006, Change Permissions for Task Scheduling, The New York Times Company;
- *http://www.ietf.org/rfc/rfc1630.txt*

Schutzbedarf anderer Systeme im gleichen Netzwerksegment gegeben ist. Zum Beispiel dann, wenn die Citrix-Lösung lediglich zum hermetisch abgeriegelten Surfen im Internet genutzt werden will. Der Server mit sämtlichen Kundendaten, der sich in der gleichen Zone befindet, ist für das Unternehmen jedoch viel wertvoller.

In einer solchen Situation kann es sich lohnen, dass das Citrix-System absichtlich in eine dedizierte Zone positioniert wird und diese nach Möglichkeiten keine zusätzlichen Kommunikationen zu anderen Zonen zulässt. Ist eine Kompromittierung des Citrix-Servers gegeben, können so wenigstens die Auswirkungen limitiert und die wahrhaft schützenswerten Assets vor unliebsamen Zu- oder Eingriffen bewahrt werden.

Zussamenfassung

Citrix erlaubt es, kostenintensive Hardware und Umgebungen für mehrere Benutzer zeitgleich verfügbar zu machen. Durch das Erweitern des gegebenen Windows-Betriebssystems wird ein wahres Multiuser-System realisiert, das theoretisch das Arbeiten von jedem vernetzten Punkt der Erde ermöglicht. Durch das Aufrufen des ICA-Clients und das Verbinden mit dem MetaFrame-Server kann komfortabel auf die altbekannte Umgebung zugegriffen werden.

Auch in Bezug auf die Sicherheit können grosse Versprechungen gemacht werden. Der zentralisierte Ansatz erlaubt es, dass ebenso die Sicherheitsmassnahmen von zentralisierter Stelle angefordert und umgesetzt werden. Vorbei sind die Zeiten, in denen ein Unternehmen von der Bereitschaft der Mitarbeiter, auf ihren Laptops die Antiviren-Lösungen auf dem neuesten Stand zu halten, abhängig ist.

Durch den zentralisierten Ansatz wird jedoch auch ein *Single Point of Failure* gegeben. Ist der Citrix-Server nicht verfügbar oder die Integrität dessen in Zweifel, kann nicht mehr ohne weiteres gearbeitet werden. Ein Angreifer braucht also nicht mehr sämtliche Laptops einer dezentralisierten Organisation anzugreifen, sondern sich lediglich dem Herzstück des Citrix-Systems anzunehmen, um einen breitflächigen Schaden zuführen zu können. Durch redundante MetaFrame-Farmen lässt sich wenigstens die Ausfallsicherheit erhöhen.

Da bei einem Citrix-Zugriff so gearbeitet werden kann, als sässe der Benutzer direkt vor dem eigentlichen Rechner, müssen die Schutzmassnahmen auch entsprechend ausgelegt werden. Netzwerkbasierte Vorrichtungen (z.B. *Firewall-* oder *Intrusion Detection-Systeme*) können nicht mehr zur Verhinderung und Entdeckung von Attacken eingesetzt werden, da sich der Anwender grundsätzlich einem legitimen und damit zugelassenen Kontext bewegt.

Im Rahmen des genutzten Betriebssystems und den durch Citrix gewährten Freigaben ist es jedoch möglich, dass er sich innerhalb des legitimen Kontexts um weitere Rechte bemühen kann. Durch verschiedene Angriffstechniken bleibt es möglich, dass sich innerhalb von Citrix weitere Anwendungen starten lassen und damit zusätzliche Zugriffsmöglichkeiten erarbeitet werden können. Von Dialogboxen über fehlerhafte Dateirechte bis hin zu hilfreichen Tastaturkombinationen kann grundsätzlich alles zu Gunsten des Angreifers ausgelegt werden.

Für Citrix-Administratoren ist es entsprechend aufwendig, die Umgebung umfassend abzusichern. Das Härten des Betriebssystems und das fortwährende Einspielen der aktuellsten *Patches* sind unerlässlich, um ein Mindestmass an Sicherheit erlangen zu können. Desweiteren müssen hostbasierte Schutzmassnahmen integriert werden, die lokale Rechteausweitungen verhindern können. Allem voran sind dies restriktive *NTFS*-Dateirechte, die dediziert und wohl überlegt appliziert werden wollen. Ein, vor allem in komplexen Umgebungen, sowohl wirtschaftlich als auch technisch schier unlösbares Problem.

Zudem können uns klassische Schutzmechanismen dabei helfen, eine Rechteausweitung im Netzwerk zu erschweren. Durch Personal Firewalls lassen sich *Loopback*-Angriffe oder ausgehende Attacken auf Systeme in naheliegenden Netzwerkzonen verhindern. Ebenso können *Intrusion Detection*-Systeme dabei helfen, entsprechende Angriffsversuche frühzeitig zu entdecken, um schnellstmöglich Gegenmassnahmen einzuleiten (z.B. Verwarnung und Ausschluss von Benutzern).

Grundsätzlich ist es wichtig, in sicherheitskritischen Umgebungen nicht auf das Heranziehen von Spezialisten zu verzichten. Eine Begleitung während des Aufbaus und das kontinuierliche Überprüfen der Installation auf mögliche Schwachstellen hilft dabei, potentielle Mängel frühzeitig ausmachen und beheben zu können [Ruef 2003]. Citrix ist und bleibt ein gewaltiges Monster, das man aber mit genügend Mut und Disziplin zu bändigen in der Lage ist. ●

Über den Autor

Marc Ruef arbeitet als Security Consultant bei der schweizer Firma scip AG und ist dort Leiter des Bereichs Security Auditing und Penetration Testing. Im Oktober 2002 ist im Data Becker Verlag sein zweites Buch mit dem Titel *Hacking Intern* erschienen. Neben einer Vielzahl von Fachpublikationen zur theoretischen Informatik und IT-Security unterstützt er diverse internationale Projekte aus diesen Bereichen. Seit 1997 betreut er die Webseite *computec.ch*, die als grösstes Archiv deutschsprachiger Publikationen zum Thema Informationssicherheit gilt. Des Weiteren ist er Entwickler des Attack Tool Kit (ATK), einem open-source Exploiting Frame-work, das als Ergänzung zu Lösungen wie Nessus generische Penetration Tests einfacher und transparenter macht. Kontakt zum Autor: marc.ruef@computec.ch *http://www.computec.ch*